

\*Subir la solución (un archivo pdf) antes del domingo a las 11:55 p.m.

\*Si los algoritmos o los programas no están debidamente identados, se les restarán puntos.

\*Para (3) subir un package en un .zip,

1.-(2.5pts)Escribir el algoritmo recursivo e iterativo de la búsqueda binaria.

2.-(2.5pts)Escribir un algoritmo de backtrack para resolver el problema de las ocho reinas.

3.-(2.5pts)Escribir un programa (y probar su resultado) que resuelva el problema de los saltos del caballo, de acuerdo a la información vista en clase y al siguiente algoritmo:

```
desplazamiento[][] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2}, {1, -2},  
{1, 2}, {2, -1}, {2, 1}}
```

```
boolean saltoCaballo(tab, fil, col, numSalto)  
    si( EstaFueraDelTablero )  
        regresa F  
    otro si( YaTieneNumeroDeSalto )  
        regresa F  
    otro si( EsElSaltoFinal )  
        tab[fil][col] = numSalto  
        regresa V  
    otro  
        tab[fil][col] = numSalto;  
        para( int[] aux: desplazamiento )  
            nf= fil+aux[0]  
            nc= col+aux[1]  
            res = saltoCaballo( tab, nf, nc, numSalto+1 )  
            si( res==V )  
                regresa V  
        fin_para  
        tab[fil][col] = 0;  
        regresa F
```

```
    fin_si
```

```
fin_metodo
```

**//Para invocar por primera vez al método recursivo e imprimir el resultado**

```
int T = 8; //Tamaño del tablero  
int[][] tab = new int[T][T]; //tablero  
int finini=0; int cini=0; //cuadro inicial
```

```
boolean resultado = saltoCaballo(tab,finini,cini,1); //Prmer salto
```

```
if (resultado) {  
    System.out.println("El resultado iniciando en "+finini+","+cini+" es:");  
    for (int fil=0; fil<tab.length; fil++) {  
        for (int col=0; col<tab.length; col++) {  
            System.out.printf("%2d ", tab[fil][col]);  
        }  
        System.out.print("\n");  
    }  
    System.out.print("\n");  
} else {  
    System.out.println("No tiene solución, seleccionar otro inicio.");  
}
```

**//Ejemplo de una solución**

El resultado iniciando en 0,0 es:

1	12	9	6	3	14	17	20
10	7	2	13	18	21	4	15
31	28	11	8	5	16	19	22
64	25	32	29	36	23	48	45
33	30	27	24	49	46	37	58
26	63	52	35	40	57	44	47
53	34	61	50	55	42	59	38
62	51	54	41	60	39	56	43

**4.-(2.5pts)** Escribir la secuencia de estados del siguiente arreglo, cuando es ordenado utilizando “quicksort”. Considerar que un estado nuevo se da cada vez que se cambia de pivote. También marcar el elemento que se utilizó como pivote.

12 9 6 3 14 17 20 10 7 2 13 18 21 4 15